

### IN THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A method of adding an event source to a transaction processing system having a workflow server engine comprising ~~the steps of~~:  
defining the event in a workflow database;  
creating at least one ~~one or more~~ executable ~~functions~~ function which creates a data structure for events coming from the event source; and  
creating a workflow to be executed on the workflow server engine in response to an event from the event source.
2. (Original) The method of claim 1, wherein the event definition includes an event id.
3. (Original) The method of claim 2, wherein the created workflow is associated with the event id so that the created workflow is executed in response to any event having the event id.
4. (Cancelled)
5. (Original) The method of claim 1, wherein the event definition includes a list of parameters associated with the event.
6. (Currently Amended) The method of claim 1, wherein the at least one ~~one or more~~ executable ~~functions~~ function is comprised of a dynamic link library.
7. (Currently Amended) The method of claim 1, wherein the at least one executable ~~functions are~~ function is designed to send an event to the workflow server engine.
8. (Original) The method of claim 1, wherein the event source is added without changing the workflow server engine.

9. (Currently Amended) The method of claim 1, further comprising ~~the step of creating~~ at least one ~~one or more rules~~ rule for associating an event from the added event source with the workflow.
10. (Original) The method of claim 9 wherein the rule includes a logic expression.
11. (Currently Amended) The method of claim 9, wherein the event definition includes at least one ~~one or more parameters~~ parameter, and wherein the created at least one rules rule includes the use of the at least one parameters parameter.
12. (Currently Amended) The method of claim 1, wherein a plurality of events are defined in the workflow database, the method further comprising ~~the step of~~ categorizing the events into a plurality of event types.
13. (Original) The method of claim 12, wherein each of the workflow server engine handles each of the event types in different ways.
14. (Currently Amended) A method of adding a new subsystem to a workflow server engine having a plurality of subsystems for providing events to the workflow server engine, the method comprising ~~the steps of~~:
- defining an event which will be generated by the new subsystem;
  - creating a dynamic link library for creating a data structure for the defined event; and
  - associating the defined event with a workflow so that the associated workflow is executed on the workflow server engine in response to an event from the new subsystem.
15. (Original) The method of claim 14, wherein the dynamic link library creates a data structure for the defined event.
16. (Currently Amended) The method of claim 14, wherein ~~the step of~~ defining the event further comprises ~~the step of~~ assigning an event id to the event.

17. (Currently Amended) The method of claim 14, wherein ~~the step of~~ defining the event further comprises ~~the step of~~ associating a plurality of parameters to the event.
18. (Original) The method of claim 17, wherein the plurality of subsystems also have a plurality of associated events.
19. (Currently Amended) The method of claim 18, further comprising ~~the step of~~ exchanging events between different subsystems during the execution of the workflow.
20. (Original) An apparatus for executing a transaction task within a transaction processing system comprising:
- a plurality of event providers for providing a source of events to the transaction processing system;
  - a database for storing information relating to the events provided by the event providers;
  - a workflow server engine for executing workflows in response to events from the plurality of event providers; and
  - a workflow editor for creating and editing workflows to be executed on the workflow server engine.
21. (Original) The apparatus of claim 20, further comprising:
- a new event provider;
  - a dynamic link library associated with the new event provider for allowing the new event provider to provide events to the workflow server engine.
22. (Original) The apparatus of claim 21, wherein the dynamic link library allows the new event provider to provide events to the workflow server engine without changing the workflow server engine.
23. (Original) The apparatus of claim 20, wherein the transaction processing system collects step execution information.

24. (Original) The apparatus of claim 20, wherein the collected information includes information relating to the number of times a branch was executed by the workflow server engine.
25. (Currently Amended) The apparatus of claim 20, wherein the collected information includes information relating to the step execution time for at least one ~~one or more steps~~ step executed by the workflow server engine.
26. (Currently Amended) A workflow execution system comprising:  
a workflow server engine adapted to execute workflow on the workflow server engine in response to an event;  
a database server;  
a plurality of subsystems for providing the events to the workflow server engine; and  
wherein components of the workflow server engine are standards-based components.
27. (Original) The system of claim 26, wherein the components are comprised of ActiveX controls.
28. (Currently Amended) A method of executing at least one ~~one or more workflows~~ workflow in a transaction processing system comprising ~~the steps of~~:  
defining a plurality of events;  
categorizing the plurality of events in to different types of events; and  
executing one of the workflows in response to at least one ~~one or more events~~ event,  
wherein different types of events are handled differently.
29. (Original) The method of claim 28, wherein the different types of events includes synchronous events in which the workflow waits for a response from a synchronous event before continuing.

30. (Original) The method of claim 28, wherein the different types of events includes asynchronous events in which a workflow being executed continues while waiting for a response to a synchronous event.
31. (Original) The method of claim 28, wherein the different types of events includes unsolicited events in which the workflow passes an unsolicited event to any workflows being executed.
32. (Currently Amended) A method of adding a new service to a transaction processing system having a workflow server engine comprising ~~the steps of~~:
- creating a configuration for the service in a database;
  - creating at least one ~~one or more~~ executable ~~functions~~ function which creates a data structure for communications coming from the service; and
  - creating a workflow to be executed on the workflow server engine in response to an event including communications with the service.
33. (Original) The method of claim 32, wherein the service is an external database access service.
34. (Original) The method of claim 32, wherein the service is a telephony service.
35. (Original) The method of claim 32, wherein the service is an email service.
36. (Currently Amended) A workflow execution system comprising:
- a centralized database server; and
  - a plurality of workflow server engines connected to the centralized database server,
- wherein the database server provides information to the workflow server engines for executing workflows in response to an event.
37. (Original) The method of claim 36, wherein the centralized database server stores workflows to be executed by the workflow server engines.

38. (Original) The method of claim 36, wherein the centralized database server stores rule sets for use by the workflow server engines.
39. (Original) The method of claim 36, further comprising a locking mechanism for preventing access to the database server.
40. (Original) The method of claim 39, wherein the locking mechanism prevents access to components of the database server while the components are being modified.
41. (Currently Amended) The apparatus of claim 20, wherein the database stores a plurality of workflows and a plurality of rules, and wherein the workflow server engine can selectively load at least one ~~one or more~~ of the workflows and rules.
42. (Currently Amended) A method of executing workflows in a transaction processing system comprising ~~the steps of~~:
- providing a database;
  - providing a workflow server engine adapted to execute workflow on the workflow server engine in response to an event;
  - storing a plurality of workflow configurations on the database; and
  - selectively loading at least one ~~one or more~~ of the workflow configurations into the workflow server engine.
43. (Currently Amended) The method of claim 42 further comprising ~~the step of~~ selectively unloading previously loaded workflow configurations.
44. (Currently Amended) The method of claim 42 wherein the workflow configurations each include at least one ~~one or more~~ executable ~~workflows~~ workflow.

45. (Currently Amended) The method of claim 42 wherein each of the at least one workflow configurations ~~each~~ include at least one ~~one or more~~ rule sets set used for selecting workflows based on events.
46. (Currently Amended) A method of handling exceptions while executing workflows in a transaction processing system comprising ~~the steps of:~~  
creating a workflow to be executed on a workflow server engine in response to an event;  
declaring at least one ~~one or more~~ exception handlers handler;  
during execution of the ~~[[a]]~~ workflow, checking for the occurrence of an exception; and  
if an exception has occurred, handling the exception according to the declaration.
47. (Original) The method of claim 46, wherein the exception handler is declared in a workflow step.
48. (Original) The method of claim 46, wherein declaration instructs the workflow to ignore the exception.
49. (Original) The method of claim 32, wherein the service is Web server request service.